Cubix

Visualizing Dynamic Networks with Matrix Cubes

— User Manual —

benjamin.bach@inria.fr
INRIA, France

September 4, 2013

This manual is a short introduction into *How to Explore dynamic networks with Cubix*.

Contents

1 Matrix Cubes **2**

2 Interface and Visual Encoding **2**
2.1 Cell Shape Encoding 2
2.2 Cell Color Encoding 3

3 Views **4**
3.1 3D View 4
3.2 Front View 5
3.3 Side View 5
3.4 Rotating Slices 5
3.5 Time Side-by-Side View 6
3.6 Vertices Side-by-side View 6
3.7 Slide Show View 6

4 View Changes and the *Cubelet*-Widget **8**

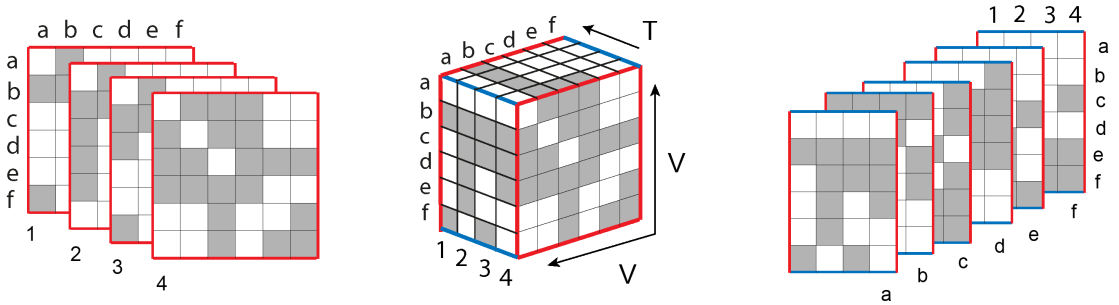
5 Filtering **8**
5.1 Lasso Selection 9
5.2 Cell Selection 9

6 Row and Column Reordering **10**

7 Controls Summary **11**
7.1 Keyboard Controls 11
7.2 Mouse Controls 11

1 Matrix Cubes

Cubix is a visualization interface for the exploration of dynamic networks with changing edge weights. The central visualization is the **Matrix Cube**, a space-time cube resulting from stacking adjacency matrices, one for each time step, in the order of time. Figure 1 shows how a matrix cube is created from the adjacency matrices.



(a) Adjacency Matrices (b) Matrix Cube (c) Vertex Slices

Figure 1: Contruction of the Matrix Cube. (a) Each time step of the network (1,2,3,4), is represented as an adjacency matrix. (b) The cube resulting from stacking those matrices. Red edges of the cube hold vertices and correspond to the rows and columns of the constituent adjacency matrices; blue edges of the cube hold time steps. (c) Slicing the cube along one of the vertex dimensions yields one *vertex slice* per vertex, showing the evolution of that vertex's ego-network.

Visualization an exploration of three-dimensional models on the screen is difficult. With Cubix, we provide transformation and decomposition operations which yield better readable 2D representations of the information contained in the cube. This manual explains what operations and visualizations Cubix supports and how they are employed.

Thoughout this manual, we use the following terminology:

- **Time Slice** — A time slice corresponds to the adjacency matrix for one time step (Figure 1(a)). A time slice shows the network's state at one time step.
- **Vertex Slice** — A vertex slice results from cutting the cube orthogonally to time slices, representing one vertex. A vertex slice resembles a table where a vertex's neighbors correspond to rows and time steps to columns (Figure 1(c)). A vertex slice shows a vertex's neighborhood over time (dynamic ego network).
- **Cell** — A cell corresponds to a connection between two vertices at one particular time.
- **Vector** — A vector corresponds to a single line of cells inside the cube; rows and columns in a single matrix correspond to vectors. In the matrix cube, vectors can also show the evolution of connectivity during time.

2 Interface and Visual Encoding

The interface of Cubix is show in Figure 2. The cube in the center shows a collaboration network. Persons (the network's vertices) are shown along the the vertical and horizontal red axes. Time is shown along the blue axes. Connections between vertices are encoded as three-dimensinal *cells* inside the cube. Network statistics are shown on the top left and the Cubelet Widget allows to switch between views on the cube (Section 4).

Cells in Cubix can vary in two ways, shape and color. Both can be set by the radio buttons on the right side of the interface.

2.1 Cell Shape Encoding

a) Edge Weight: By default, cell size indicates edge weight in a logarithmic way. Larger cells mean higher edge weight, smaller cells lower edge weight. In this manual edge weight indicates the number

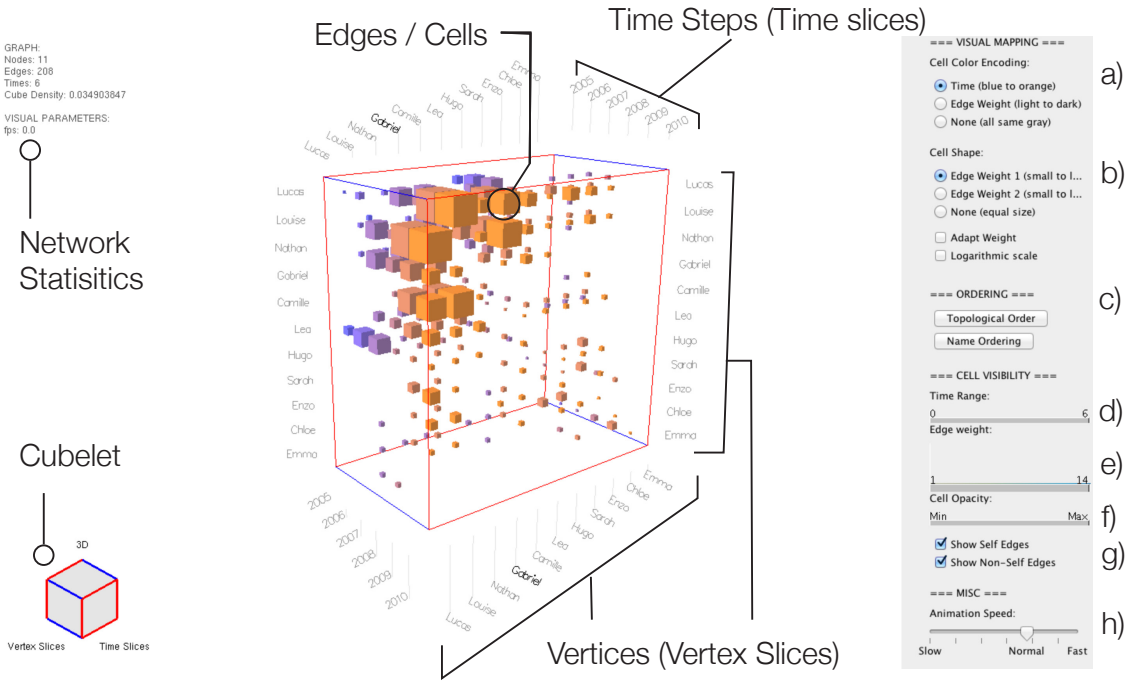


Figure 2: Cubix user interface. a) Cell color settings, b) Cell shape settings, c) Row and column ordering, d) Time range slider to set visible time slices, e) Edge weight slider to define visiblity of cells depending on edge weight. A histogram indicates distribution of edge weight. g) Settings to show or hide self and/or non-self edges, h) Slider to define animation speed.

of co-publications per year.

b) Edge Weight 2: When cells are the visual space between two cells increases, which can make it hard to relate cells between the same two nodes over time. This second encoding shows a cells with equal length, making all edges between the same two nodes one single visual vector over time Figure 3.

c) None: No shape encoding means that all cells have equal size.

(a) Edge Weight 1 (b) Edge Weight 2

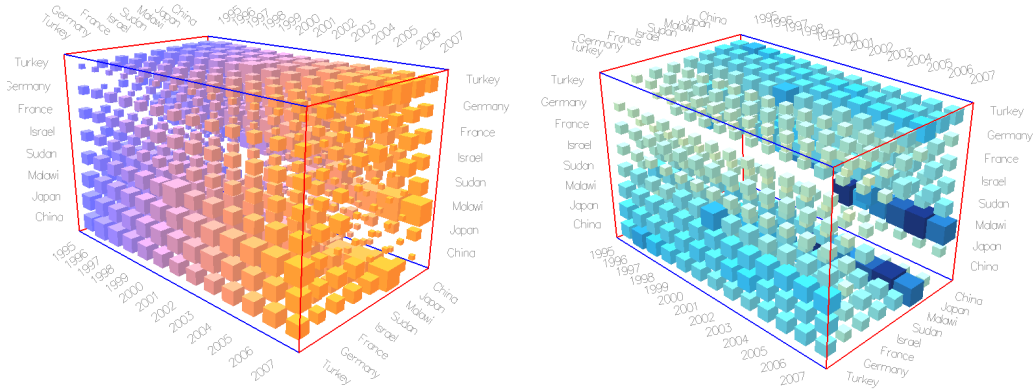
Figure 3: Difference between Edge Shape Encoding seen in Side View. Edge Weight 2 connects the cells of the same node pair, leading to visual continuity.

2.2 Cell Color Encoding

For coloring cells, three modes exist, set by the radio buttons on the right side in the interface.

a) Time Encoding colors cells according to which time slice they belong to (Figure 4(a)). The color scale ranges from blue (early times), via purple and indigo (middle time steps) to orange (recent times).

b) Weight Encoding colors cells according to their weight, ranging from light turquise (low weight) to dark blue (high weight) (Figure 5(b)). Weight encoding makes heavy edges stick out.

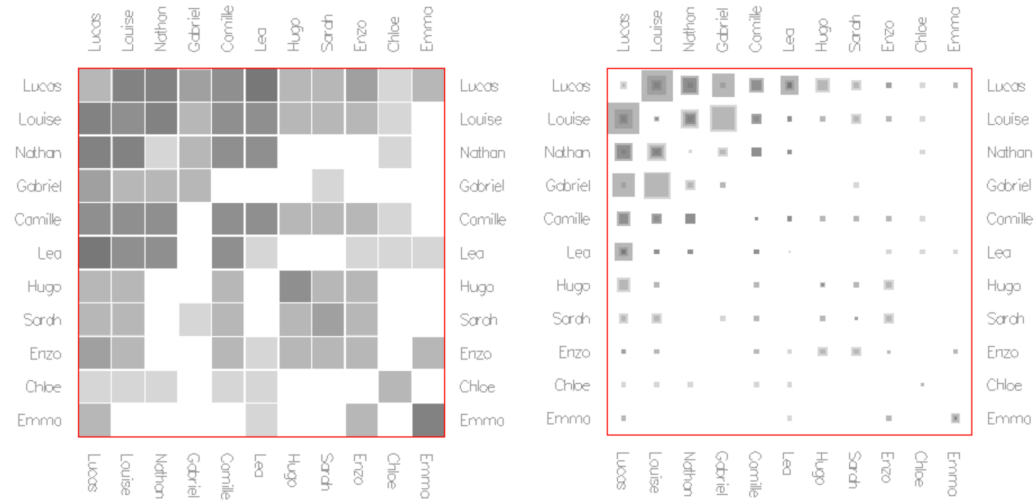


(a) Time encoding (blue to orange)

(b) Value encoding (light to dark)

Figure 4: Cell encoding in Cubix

c) No Encoding shows all cells in the same dark gray. When applying opacity to cells and seeing the cube from the Front, node pairs with many connections over time stick out by appearing darker (Figure 5). More on the Front view and other views in Section 3.



(a) Quantity of connection between node pairs over time

(b) Edge weight encoded in cell size.

Figure 5: No cell color encoding can highlight quantity of connections between nodes over time.

3 Views

Various views can be derived from the matrix cube by applying a combination of natural operations to it: rotation, projection, slicing, filtering, layout and flip-through.

3.1 3D View

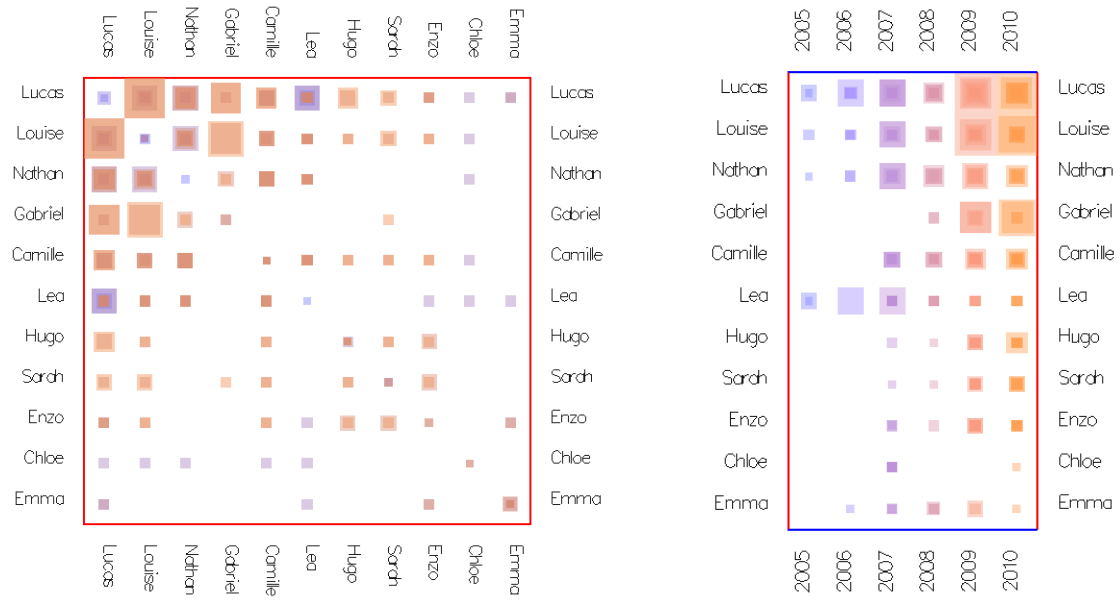
The 3D view, as shown in Figure 2, view can be

- **rotated** — move mouse while left button pressed
- **panned** — move mouse while right button pressed
- **zoomed** — scroll mouse wheel, and

When hovering cells, the corresponding labels for vertices and times get highlighted to make their identification easier. Another label indicates the edge weight.

3.2 Front View

Figure 6(a) shows the cube, rotated and projected so that it shows the adjacency matrix with all *time slices* superimposed using alpha blending. Cells inside the cube become translucent, and provide a summary of the connections between any two vertices. Large cells indicate high edge weight. Dark cells, due to super imposition, indicate frequent connections. Topological clusters emerge from reordering rows and columns. Illustrated in Figure 6(a), *Luise* and *Lucas* collaborate frequently, while *Nathan* and *Lucas* collaborated in a few years only, but on many articles. Using to time encoding, clusters and connections can be roughly situated in time, e.g. Lucas' individual publications (Lucas $\times$ Lucas) are much older than his collaborations.



(a) Front View of Matrix Cube.

(b) Side view of Matrix Cube.

Figure 6: Projections on the cube

3.3 Side View

When rotating the cube to its left side face, a projection as shown in Figure 6(b) is obtained. Rows still show vertices, but columns represent time, one column per time step. Time runs from left to right. Cells in this view summarize all connections of a vertex over time steps. The side view makes it easy to spot the overall evolution of each vertex's degree, and to identify time steps (years in our example) that feature denser or sparser networks. In Figure 6(b) for example, the year 2008 shows less publications than 2007, because the corresponding column contains smaller cells.

3.4 Rotating Slices

To individually observe an individual slice, when in Front or Side view, you can rotate individual time slices (in the *side view*) or vertex slices (in the *front view*) projection. To rotate a slice, click on the corresponding label. Rotating a slice can be compared to rotating a single book on a book shelf; after rotation, the slice can be observed individually while context of the current view is preserved. This view is useful when an interesting pattern in the front or side view has been found and one particular slice requires particular investigation. Note that you can rotate as many slices as you want. Figure 7

shows the rotation of a time slice from the view in Figure 6(b). Slices can be rotated back to its initial position by clicking on any cell within that slice.

Figure 7: Focus and context view of a time slice. (a) Time $\times$ vertices projection, (b) after rotating time slice for 2009.

3.5 Time Side-by-Side View

A common technique to show spatio-temporal data is to use small pictures side by side, one for every time step. Since time in the matrix cube is discrete, decomposition is straightforward. In cubix, time slices (matrices) can be juxtaposed, allowing for pairwise comparison and individual analysis (Figure 8). Hovering a cell highlights all connections between the same node pair over time.

3.6 Vertices Side-by-side View

Figure 9 shows all *vertex-slices* laid out side by side. In this example, cell color was already mapped to cell size to allow for better cross-slice comparison. Each vertex-slice shows the dynamic ego-network for each vertex, enabling the comparison of individual connection patterns across vertices of the network. Note that pan and zoom is enabled in both side-by-side representations.

3.7 Slide Show View

Yet another way to explore individual slices is to present them one at a time, such as in a flip-book or a slide show. From the 3D view, users can select any single vertex or time slice, consequently hiding

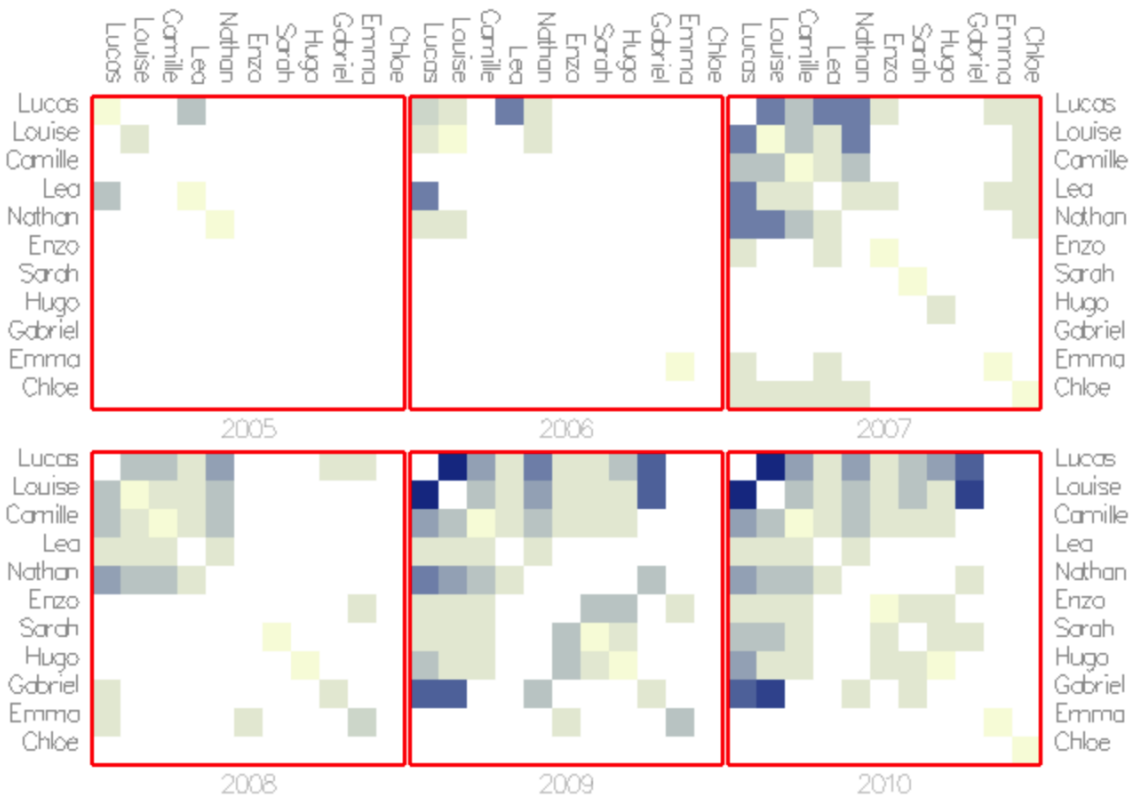


Figure 8: All time slices juxtaposed using value encoding and equal cell size. Darker cells indicate more edge weight. However switching between cell encodings must be done manually by the user.

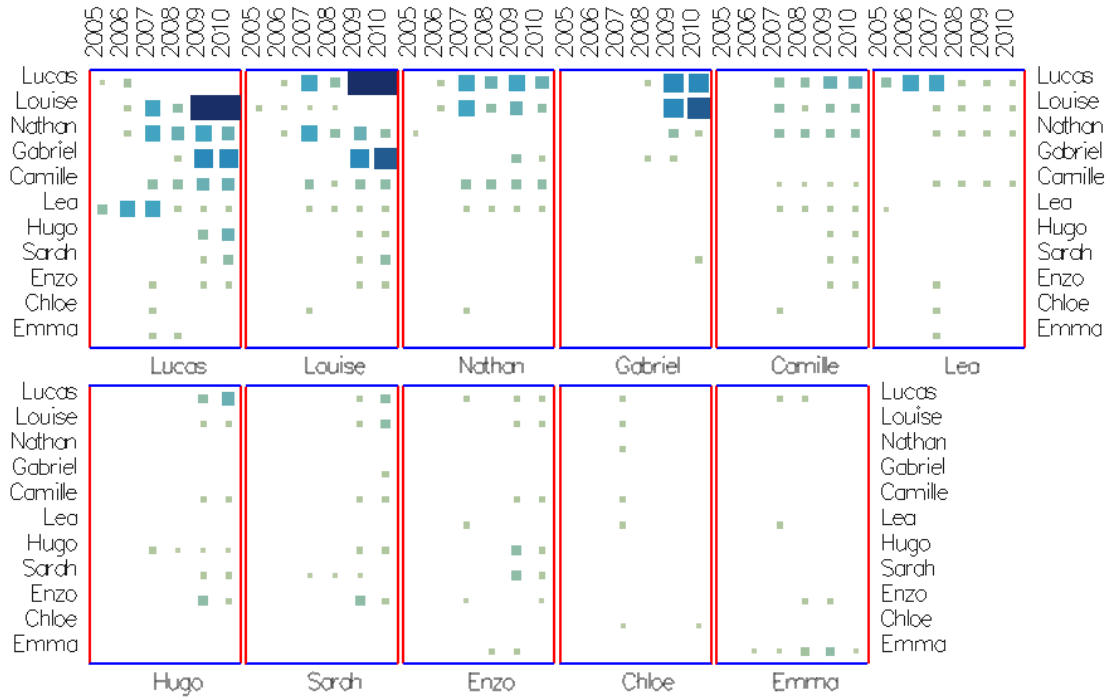
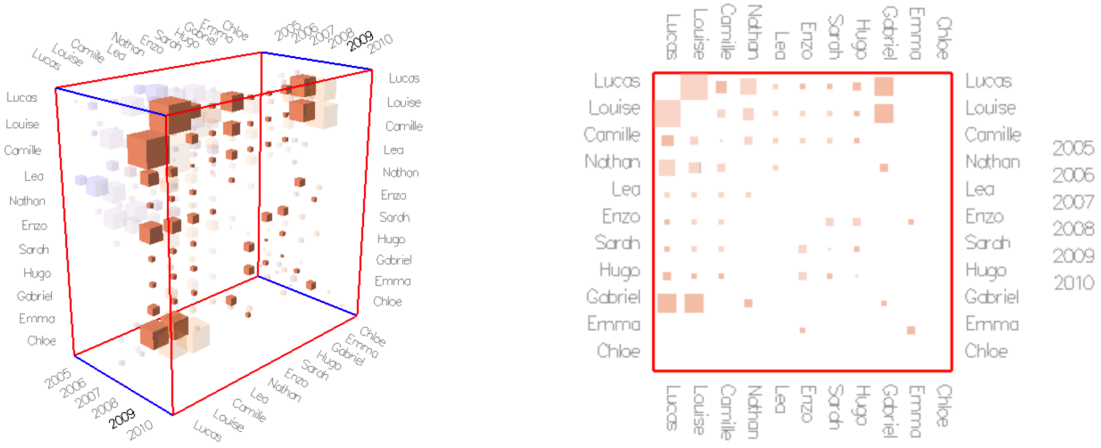


Figure 9: Small multiples of vertex slices using value cell encoding. Cells are colored using value encoding to highlight differences in weight across slices.

all other slices. Users can then switch to the front or side view and obtain a projection that shows only the selected slice (Figure 10).

- Using the *up* and *down* arrow keys switches to the next or previous **time slice**.
- Using the *left* and *right* arrow keys switches to the next or previous **vertex slice**.



(a) (b)

Figure 10: Selecting a single slice in the 3D view and rotating the cube shows the selected slice only.

4 View Changes and the *Cubelet*-Widget

For quick changes between the previously explained views, Cubix privides number shortcuts (1-5), or alternatively the Cubelet widget shown in Figure 11(a). The cubelet is a stylized representation of a Matrix Cube with sensitive surfaces. Surfaces can be clicked in order to switch to the associated view. Additionally, the Cubelet reflects the current view. Number shortcuts and Cubelet interaction is as follows:

1. **3D View** — Click on top face of the cubelet. The entire cubelet becomes gray to indicate that all faces are visible.
2. **Front View** — Click on right side of cubelet. The front face gets becomes to indicate the front projection.
3. **Side View** — Click on left side of cubelet. The side face gets gray to indicate the side projection.
4. **Time Side-by-Side** — Drag mouse on right side of cubelet as to pull it apart. Results in the image show in in Figure 11(b).
5. **Vertices Side-by-Side** — Drag on left side of cubelet as to pull it apart. Results in the image show in in Figure 11(c).

All view switches are indicated though animated transitions. Animated transitions can be shortened by holding **shift**, and fastened with the **alt** key.

5 Filtering

Independently from the current view, visibility of cells can be adapted by different filtering methods, explained in the following. The corresponding sliders can be found on the bottom of the screen (Figure **??**).

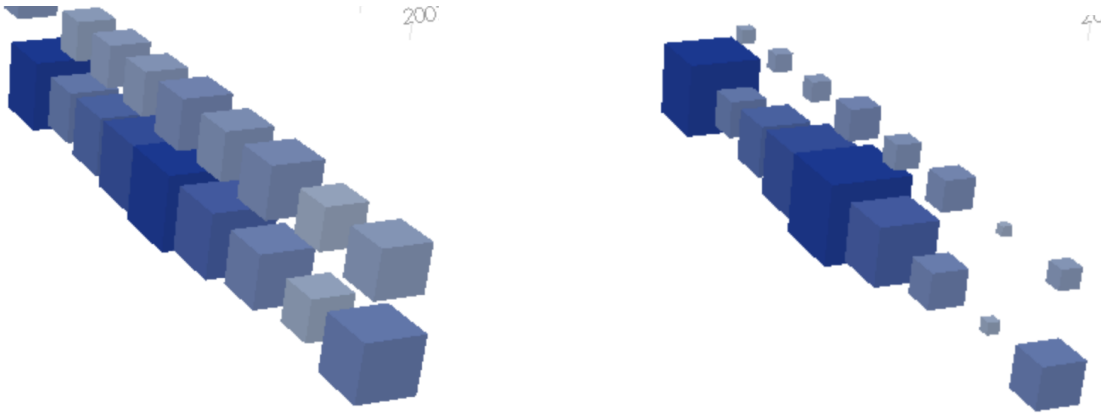
- **Opacity** — The right side of the *opacity range slider* sets general cell opacity. The left side sets opacity of cells which are a) not in the current time range or b) not in slices currently selected.

(a) (b) (c)

Figure 11: Different states of the Cubelet widget indicating the current view of the matrix cube. (a) front face is shaded and selected slices highlighted, (b) slicing along the time dimension with selected slice highlighted, and (c) slicing along the vertex dimension.

- **Weight** — The weight range slider allows to see only cells of a particular edge weight range. When the **ADAPT WEIGHT**-button is active, the remaining cubes are scaled so that their size varies according to the currently visible wight range. This means that cells at the lower range of visible weight are very small, while cells on the higher end have maximal size. Such an adapted weight range allows to better perceive the weight distribution within a given weight range (Figure 12).
- **Time** — The weight range slider allows to see only cells of a particular time range.
- **Nodes** — Similiar to time filtering, node slices can be selected by clicking on the corresponding labels on the cube. Non-selected slices are rendered translucent. The examples in Figure 13 have been created by node filtering and show the evolution of weight of two edges over time.

Changes in opacity and time or node selection are independent from each other and remain consistent across view changes.



(a) No weight adaption (b) Weight adaption

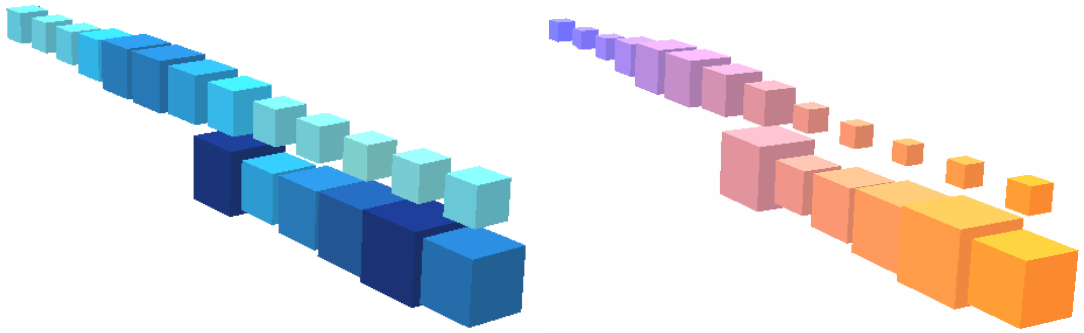
Figure 12: Weight adaption scales cube only within the currently visible weight range.

5.1 Lasso Selection

Holding the ALT-key while dragging the mouse activates a lasso selection tool which allows to circle cells. After releasing the mouse, only the selected cells remain visible throughout all views. Clicking somewhere inside the cube while holding the ALT key deactivates the lasso selection.

5.2 Cell Selection

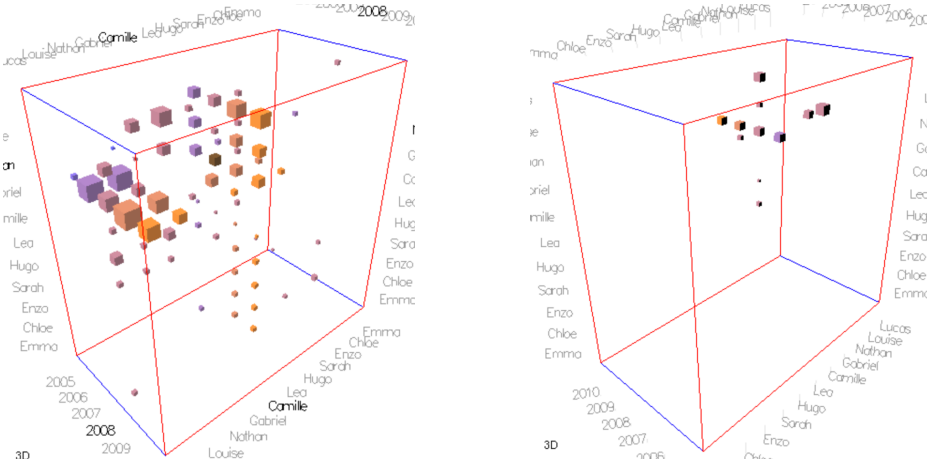
Cells can be selected to highlight their context. When the cell is clicked once, only the three slices, which the cell is the intersection of, remain visible (Figure 14(a)). The front and side views show only the time and vertex slice respectively, allowing to investigate the cell's context.



(a) Value encoding (b) Time encoding

Figure 13: Two edge vectors in value and time encoding.

Clicking the selected cell twice remains only the vectors, which the cell is the intersection of, visible (Figure 14(b)).



(a) Slices of selected node (b) Vectors of selected node

Figure 14: Selecting a cell can show slices or vectors this cell is part of. The selected cells becomes brown.

6 Row and Column Reordering

The order or rows and columns in the cube is determined by an algorithm that optimizes patterns in the matrices. Across all matrices in the cube, there is only one ordering of rows and columns possible. Hence, the more time slices exist and the more they differ, the less efficient an optimization can be. To optimize reordering for one time slice or a subset of time slices you can select individual time slices, e.g. with the time slider, or filter cells with the edge weight slider, and re-run the reordering (**RE-ORDER**-button). All time slices in the cube will consequently obtain the same ordering, obtained from the visible cells. To facilitate the look-up for vertex names, Cubix provides the **NAME-ORDERING**-button which orders vertices in an alphanumerical way. Ordering remains consistent across view changes.

7 Controls Summary

7.1 Keyboard Controls

- Key **1** - 3D Cube View
- Key **2** - Front View
- Key **3** - Side View
- Key **4** - Time Side-by-Side
- Key **5** - Vertices Side-by-Side
- **Shift** - Slow View transitions
- **Alt** - Fast view transitions
- **Arrow Up** - Move horizontal vertex slice selection in the cube up
- **Arrow Down** - Move horizontal vertex slice selection in the cube down.
- **Arrow Left** - Move vertical vertex slice selection in the cube to the left.
- **Arrow Right** - Move vertical vertex slices selection in the cube to the right.

7.2 Mouse Controls

- **Drag mouse + left button** - Rotate cube
- **Drag mouse + right button** - Pan
- **Drag mouse + left button + Alt** - Create lasso selection
- **Click left mouse button + Alt** - Remove lasso selection